Supplementary Material

He Wang[1] Srinath Sridhar[1] Jingwei Huang[1] Julien Valentin[2]
Shuran Song[3] Leonidas J. Guibas[1,4]
[1]Stanford University [2]Google Inc. [3]Princeton University [4]Facebook AI Research

Abstract

In this supplementary material, we provide additional details and results for the method proposed in the paper.

1. Implementation and Computation Times

Our network is implemented on Python 3, Keras and Tensorflow. The code is based on MatterPort's Mask RCNN implementation[1]. The network uses Feature Pyramid Network (FPN)[4] and a ResNet50 backbone[3].

Our network takes images with a resolution of 640×360 as input. We achieve an interactive rate of around 4 fps on an Intel Xeon Gold 5122 CPU @ 3.60GHz desktop with a NVIDIA TITAN Xp. Our implementation takes an average time of 210 ms for neural network inference and 34 ms for pose alignment using Umeyama algorithm.

2. Scanned Real Instances

Our real dataset contains 6 object categories and 42 real scanned unique instances. For each category, we collect 7 instances with 4 for training and validation and the rest 3 for test. Figure 1 show a subset of our instances where one can see a large intra-category shape variance in the dataset. The first row are instances used in training. The second and third rows are held-out instances for testing.

3. Result Visualization

Here we provide more visual results of the 6D pose and size estimation. Due to sufficient training data, our method achieves very promising performance on CAMERA25 validation set as shown in Figure2. On REAL275 test set, we still observe decent performance even though the amount of real training data is small. We observe several failure modes on real data, including missing detection, wrong classification, and inconsistency in predicted coordinate maps.

4. Comparisons on the OccludedLINEMOD Dataset

The comparison between our method and other existing methods with 2D projection metric on OccludedLINEMOD dataset[2] is shown in Figure 4.

References

[1] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 1

[2] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3364–3372, 2016. 1, 3

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 1

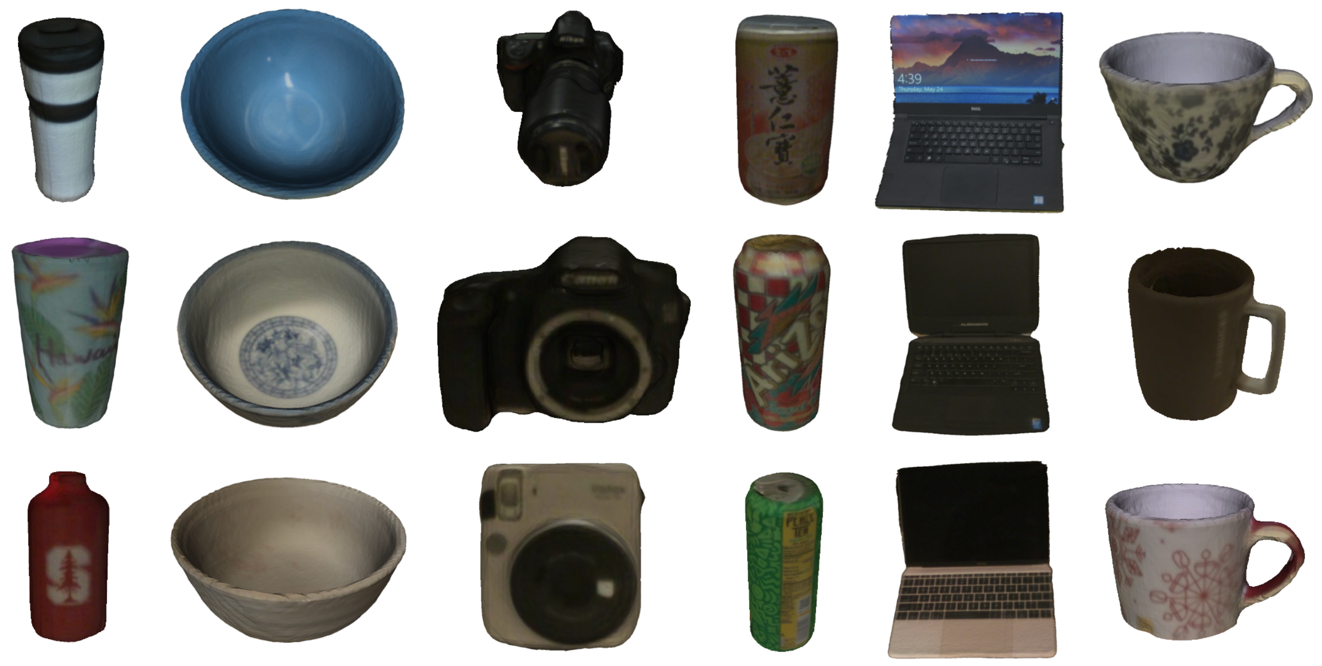

Figure 1. A subset of our 42 instances in our real dataset. From the left to the right column, they are: bottle, bowl, camera, can, laptop and mug. The first row shows instances that belong to training set. The second and third rows show instances that belong to test set.

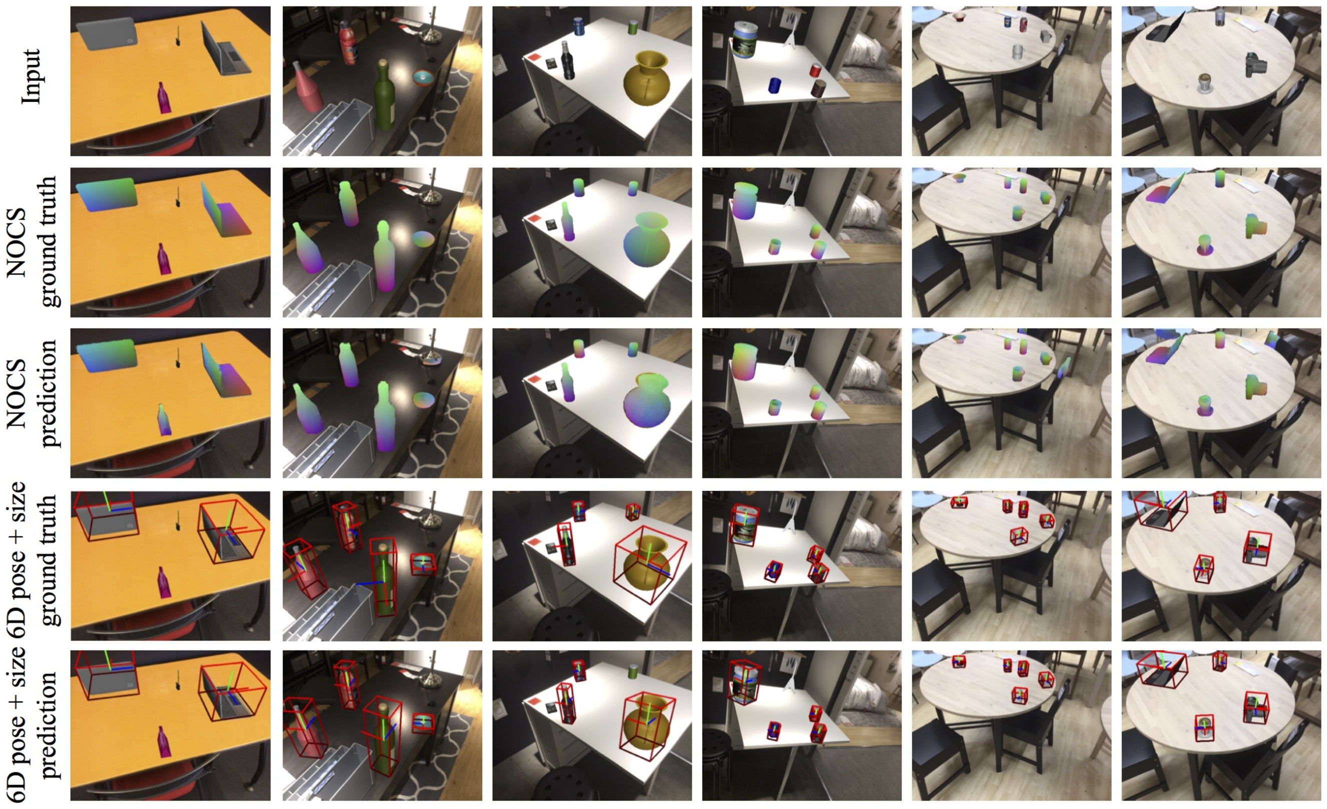

Figure 2. Qualitative result on CAMERA25 validationset set. The top row shows the input rgb image. The second and the third rows show the color-coded ground truth and predicted NOCS maps. The fourth and the fifth rows show the ground truth and predicted 6D pose (axis) and size estimation (red tight bounding box).

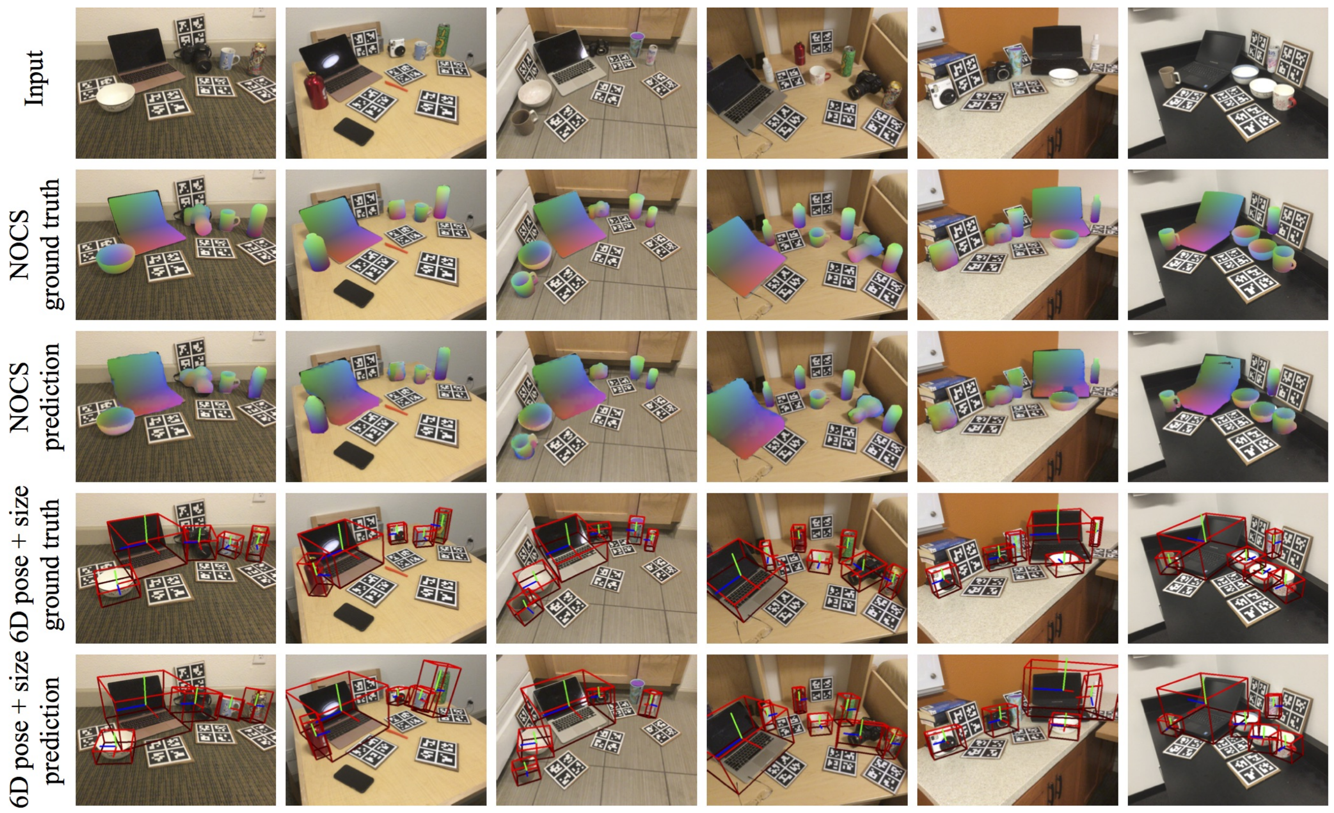

Figure 3. Qualitative result on REAL275 test set.

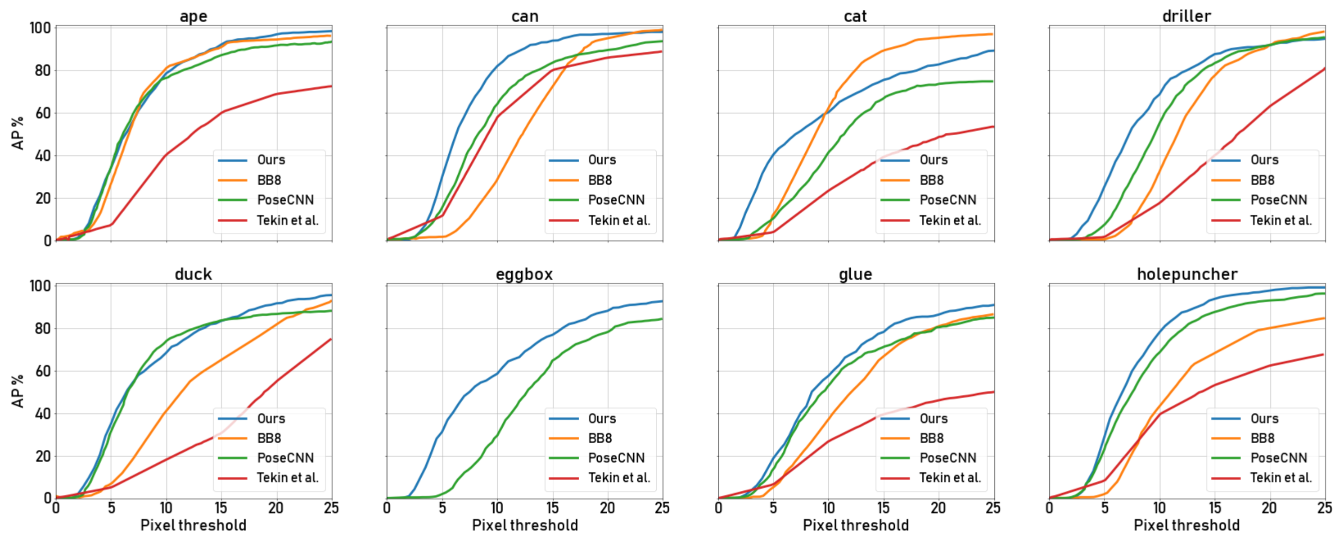

Figure 4. Comparison with state-of-the-art RGB or RGB-D based methods on the OccludedLINEMOD dataset[2].